AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A computer implemented apparatus ~~system~~ that facilitates generalized comprehension in an imperative language, comprising:

a computer readable medium comprising:

a language component for causing a computer to receive ~~that receives~~ a user input expression comprising comprehension notations by specifying a user-definable type or class as part of a generalized comprehension that appears as an outside operator to a fixed or static comprehension, enabling programming of comprehension notations in the imperative language;

an interface component for causing the computer to describe ~~that describes~~ a meaning of the comprehension by associating the user-definable type or class with an interface that implements the class, defining methods for the interface in order to supply desired functionality for the class, and by applying the class and associated interface in an exterior manner to a list or fixed comprehension function; and

a translation component for causing the computer to analyze ~~that analyzes~~ the meaning of the comprehension notations and facilitates execution of the comprehension notations in accordance with the imperative language.

2. (Canceled).

3. (Canceled).

4. (Currently Amended) The apparatus ~~system~~ of claim 1, the translation component includes at least one of just-in-time compilation techniques, interpretive techniques, and source code compilation techniques.

5. (Currently Amended) The apparatus ~~system~~ of claim 1 [[2]], the language component enables users to define at least one of an implicit expression, an explicit expression, a mathematical expression, a database expression, and a processing expression in accordance with the generalized comprehension.

6. (Currently amended) The apparatus ~~system~~ of claim 1, the comprehension notations comprise a mathematical parameterized monad.

7. (Canceled).

8. (Currently Amended) The apparatus ~~system~~ of claim 1, wherein the interface component is associated with at least one of a final results function, an accumulation function for intermediate results, an early termination function, and a default value.

9. (Currently Amended) The apparatus ~~system~~ of claim 1, wherein the language, interface and translation components operate upon a generalized comprehension further comprising an assignment expression or a yield return statement.

10. (Canceled).

11. (Currently Amended) The apparatus ~~system~~ of claim 1, wherein the language, interface and translation components operate upon a generalized comprehension further comprising a relational database expression.

12. (Currently Amended) The apparatus ~~system~~ of claim 11, wherein the relational database expression is employed for a query of a database.

13. (Canceled).

14. (Canceled)

15. (Canceled).

16. (Currently amended) A method for providing a programming environment for a user to input a generalized comprehension that is automatically translated into a language form in an imperative language, comprising:

defining a list comprehension expression; ~~and~~

defining a generalized comprehension class as an exterior component to the list comprehension expression within an imperative language environment;

providing an interface class for the generalized comprehension class.

17. (Canceled).

18. (Original) The method of claim 16, further comprising defining a results function for the interface class.

19. (Original) The method of claim 18, the results function returns a type that is at least one of similar and dissimilar to a type associated with the generalized comprehension class.

20. (Original) The method of claim 16, further comprising compiling the list comprehension expression and the generalized comprehension class to produce an executable format for the imperative language environment.

21. (Original) The method of claim 16, further comprising defining at least one relational database expression.

22. (Currently amended) The computer-implemented apparatus of claim 1, wherein the [[A]] computer readable medium has ~~having~~ a data structure stored thereon, comprising:

a first data field that defines **[[a]]** the static comprehension notation in the imperative language;

a second data field that defines **[[a]]** the generalized comprehension notation received from a user input and comprising the static comprehension notation and a comprehension notation external to the static comprehension notation; and

a third data field that links the static comprehension notation with the generalized comprehension notation.

23. (Previously Presented) The medium of claim 22, wherein the data structure further comprises an interface field associated with the comprehension notation and at least one method associated with the interface field.

24. (Previously Presented) The medium of claim 22, wherein the interface field is associated with at least one of a final results function, an accumulation function for intermediate results, an early termination function, and a default value.

25. (Previously Presented) The medium of claim 22, wherein the generalized comprehension notation is associated with a user-defined expression.

26. (Canceled).

27. (Canceled).

28. (Original) The medium of claim 22, further comprising a comprehension type for a direct aggregation of collections.

29. (Original) The medium of claim 22, further comprising a field for an evaluation that is deduced from analyzing a portion of a collection.

30. (Original) The medium of claim 22, further comprising a field associated with at least one of a default value and an initialization value.

31. (Original) The medium of claim 22, further comprising a comprehension type that implicitly implements an interface.

32. (Original) The medium of claim 22, further comprising an interface pattern for defining aggregation functions on collections.

33. (Canceled).

34. (Canceled).

35. (New) A method for providing a programming environment for a user to input a generalized comprehension that is automatically translated into a language form in an imperative language, comprising:

specifying a user-definable type or class as part of a generalized comprehension that appears as an outside operator to a fixed or static comprehension;

associating the user-definable type or class with an interface that implements the class;

defining methods for the interface in order to supply desired functionality for the class;

applying the class and associated interface in an exterior manner to a list or fixed comprehension function;

defining a results function to receive output from the class; and

compiling the generalized and associated list comprehensions with the interface methods to produce an executable format.